Product Requirements Document: Code-Only Design System Builder (V1)

Author: Manus AI **Date:** June 18, 2025 **Version:** 1.0

1. Introduction

This document outlines the product requirements for Version 1 (V1) of the Code-Only Design System Builder within the Ferdinand platform. Ferdinand is a comprehensive brand management platform, and this new feature aims to empower users to create, manage, and export their brand's design system directly within the application. V1 will focus on providing a robust code-centric solution, allowing users to define and preview their design tokens (colors, typography, spacing, etc.) in real-time and export them for use in web development projects. Integration with Figma is out of scope for V1 but planned for future iterations.

The primary goal is to provide users with a tangible tool to codify their brand's visual identity, ensuring consistency and accelerating the design-to-development workflow. The provided raw and semantic tokens will serve as the foundational starting point for this system, with the ability for users to build upon them.

2. Goals

The overarching goals for the Code-Only Design System Builder (V1) are:

- **Empower Users:** Provide Ferdinand users with a powerful, intuitive tool to define and manage their brand's design system directly within the application.
- **Ensure Consistency:** Facilitate the creation of a consistent visual identity across all digital touchpoints by centralizing design token management.
- **Accelerate Development:** Streamline the design-to-development handoff process by enabling direct export of design tokens in developer-friendly formats.
- **Real-time Feedback:** Offer immediate visual feedback on design system changes through a real-time preview mechanism.
- **Foundational Building Block:** Establish a solid foundation for future enhancements, including potential integrations with design tools like Figma.

3. Scope

3.1. In-Scope for V1

- **Design System Definition:** Users can define and manage core design system elements, including:
  - **Color Palettes:** Brand, Neutral (Gray Scale), and Interactive colors, based on the provided raw and semantic token structure.
  - **Typography System:** Font families, type scales (responsive), line heights, and letter spacing, utilizing the provided raw and semantic tokens.
  - **Spacing & Layout:** Tokenized spacing units and grid breakpoints.
  - **Interactive States:** Definition of tokens for hover, active, clicked, visited, and inactive states.
  - **Components (Basic Definition):** Ability to define basic properties for initial components like Buttons, Form inputs, Cards, Alerts, Overlays + modals, and Tabs. This will primarily involve linking to existing tokens rather than building full component libraries within the builder.
- **Input Form:** A dedicated section within the "Design System" tab in Ferdinand will provide an input form for users to modify design system tokens and properties.
- **Real-time Preview:** A dynamic preview area will display dummy content (e.g., a landing page with client's logo) that updates in real-time as users make changes to the design system tokens. The client's logo will be pulled from existing client data.
- **Export Capabilities:** Users can export their defined design system tokens in the following formats:
  - **Tailwind CSS Configuration:** A configuration file compatible with Tailwind CSS.
  - **General CSS:** A standard CSS file with custom properties (CSS variables).
  - **SCSS:** An SCSS file with variables.
- **Token Management:** The system will utilize and build upon the existing raw and semantic token structure provided by the user as the starting point.
- **Security:** The feature will operate within Ferdinand's existing authentication and authorization framework, ensuring it is accessible only to logged-in users with appropriate permissions.

3.2. Out-of-Scope for V1

- **Figma Integration:** Direct integration with Figma (via REST API or Plugin API) for two-way synchronization, conflict resolution, or design system validation is explicitly out of scope for V1. This will be addressed in subsequent phases.
- **Iconography Management:** While planned for the future, comprehensive icon management and synchronization are not part of V1.

- **Advanced Component Library Building:** The V1 will not include a full-fledged component builder with visual drag-and-drop interfaces or complex component property management beyond linking to existing tokens.
- **Version Control for Design System:** Advanced versioning or history tracking for design system changes within the builder is not part of V1.
- **Automated Conflict Resolution:** Since Figma integration is out of scope, conflict resolution mechanisms related to external design tools are not applicable for V1.
- **Real-time Sync with External Tools:** Real-time synchronization with any external design tools or codebases is out of scope for V1.

4. User Stories & Acceptance Criteria

This section details the key user interactions with the Code-Only Design System Builder and the criteria for successful implementation.

4.1. User Stories

- **As a designer/brand manager, I want to access the Design System Builder** so that I can define and manage my brand's visual identity within Ferdinand.
- **As a designer/brand manager, I want to see a real-time preview of my design system changes** so that I can immediately visualize the impact of my adjustments on dummy content.
- **As a designer/brand manager, I want to modify design tokens (colors, typography, spacing, interactive states)** through an intuitive input form so that I can customize my brand's visual elements.
- **As a designer/brand manager, I want to see my client's logo and brand colors applied to the real-time preview** so that I can visualize the design system in the context of their specific brand.
- **As a designer/brand manager, I want to save my design system changes** so that my defined brand guidelines are persisted within Ferdinand.
- **As a developer, I want to export the design system as Tailwind CSS configuration** so that I can easily integrate it into my Tailwind-based web projects.
- **As a developer, I want to export the design system as general CSS** so that I can use it in any web project via CSS custom properties.
- **As a developer, I want to export the design system as SCSS** so that I can leverage it in my SCSS-based projects.

4.2. Acceptance Criteria

- **Access to Builder:**
  - The "Design System" tab is visible in the client sidebar when the `figma_integration` feature toggle is enabled.
  - Clicking the "Design System" tab navigates to the Design System Builder page.
  - The Design System Builder page loads without errors and displays the input form and real-time preview area.
- **Real-time Preview:**
  - The preview area displays dummy content (e.g., a sample landing page layout).
  - The client's logo (pulled from existing data) is displayed prominently in the preview.
  - The preview dynamically updates to reflect changes made in the input form for colors, typography, spacing, and interactive states within 1-2 seconds of input change.
  - The preview accurately renders the defined raw and semantic tokens.
- **Design Token Modification:**
  - The input form provides fields for all in-scope design system elements (colors, typography, spacing, interactive states).
  - Input fields are user-friendly (e.g., color pickers for colors, dropdowns for font families, numerical inputs for spacing).
  - Changes made in the input form are immediately reflected in the real-time preview.
  - The form allows for scrolling, enabling users to view and modify all sections of the design system.
- **Saving Changes:**
  - A "Save" button is present and clearly visible.
  - Clicking the "Save" button persists the current design system configuration to the backend.
  - A success message is displayed upon successful save.
- **Export Capabilities:**
  - Dedicated buttons or options are available for exporting to Tailwind CSS, General CSS, and SCSS.
  - Clicking an export option generates a downloadable file in the specified format.
  - The exported files accurately reflect the currently saved design system tokens.

    - **Tailwind CSS Export:** The generated file is a valid Tailwind CSS configuration file, mapping semantic tokens to Tailwind classes or custom properties where appropriate.
    - **General CSS Export:** The generated file contains CSS custom properties (variables) for all defined tokens, following a clear naming convention.
    - **SCSS Export:** The generated file contains SCSS variables for all defined tokens, following a clear naming convention.

5. Technical Considerations

5.1. Existing Infrastructure Leverage

- **Frontend:** Utilize React 18 with TypeScript, Vite, Tailwind CSS + Radix UI primitives (shadcn/ui), TanStack React Query, React Context, Wouter, and React Hook Form with Zod validation.
- **Backend:** Node.js with Express.js, PostgreSQL with Drizzle ORM.
- **Design System API:** The existing `/api/design-system` endpoint should be extended or utilized to handle the reading and writing of the design system configuration, including the raw and semantic tokens.
- **Color System:** Integrate with the existing color system for palette management.
- **Typography System:** Integrate with the existing typography system for font management and type scales.
- **Logo System:** Leverage the existing logo manager to pull the client's logo for the real-time preview.
- **File Conversion System:** While not directly used for V1, be mindful of its capabilities for future export needs.

5.2. Data Model

- The design system configuration should be stored in a structured format (e.g., JSON) that aligns with the existing `theme.json` and raw tokens structure. This should include:
    - Color definitions (brand, neutral, interactive) with their raw and semantic mappings.
    - Typography definitions (font families, type scales, line heights, letter spacing) with raw and semantic mappings.
    - Spacing definitions with raw and semantic mappings.
    - Interactive state definitions.
    - Basic component property definitions (linking to tokens).

5.3. Performance

- The real-time preview must be highly performant, with minimal latency between input changes and visual updates (target: 1-2 seconds).
- Saving and export operations should be efficient and not block the UI for extended periods.
- Optimize data fetching and updates to ensure a smooth user experience.

5.4. Security

- All interactions with the Design System Builder must be authenticated and authorized through Ferdinand's existing security mechanisms.
- Ensure proper input validation and sanitization to prevent security vulnerabilities.
- Data stored and transmitted must adhere to Ferdinand's security best practices.

6. Deployment & Rollout

- The feature will be deployed as part of a regular Ferdinand release cycle.
- Initial rollout may target a subset of users or internal testing before a broader release.
- The feature toggle system (`figma_integration`) will be used to control the visibility of the new Design System tab.

7. Future Considerations (Beyond V1)

- **Figma Integration:** Implement two-way synchronization with Figma, including conflict resolution.
- **Iconography Management:** Integrate a comprehensive icon management system.
- **Advanced Component Builder:** Develop more advanced component building capabilities within the application.
- **Version Control:** Implement versioning and history tracking for design system changes.
- **Collaboration Features:** Explore features for collaborative design system management.